Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2017, 4(9):24-30**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Artificial Bee Colony Algorithm based on Novel Search Strategy

Hui-ming XIA, Zhi-gang WANG

Nanjing Normal University Taizhou College, Taizhou 225300, China

Abstract Artificial bee colony algorithm is an optimization algorithm inspired by the behavior of honey bees. The algorithm has been used to solve many kinds of numerical function optimization problems. It performs well in most cases, however, the solution search equation of artificial bee colony algorithm exist some disadvantages when solving complex functions, such as the convergence speed is not fast enough, easy to fall into local optimum. In order to improve the algorithm performance, we propose two novel solution search equations. In our method, the employed bees search only around the random solution to improve exploration, and the onlooker bees search only around the solution which is selected from the population depending on the roulette wheel to improve exploitation, at the same time, we use a more robust calculation to determine and compare the quality of alternative solutions. Experiments are conducted on a set of 12 benchmark functions, and the results demonstrate that the new algorithm has fast convergence and high accuracy than several other ABC-based algorithms.

Keywords Artificial bee colony algorithm, Swarm intelligence, Search equation

1. Introduction

The Artificial Bee Colony (ABC) algorithm proposed by Karaboga [1] is a new evolutionary algorithm based on swarm intelligence. The performance of ABC algorithm has been compared with other evolutionary algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE). The comparison were made based on various numerical benchmark functions, and the comparison results demonstrated that ABC algorithm can produce a more optimal solution [2-4]. Due to its simplicity and ease of implementation, ABC algorithm has been applied to solve many practical optimization problems [5-7] since its invention in 2005.

However, the traditional search equation of artificial bee colony algorithm exist some disadvantages when solving complex functions, such as the convergence speed is not fast enough, easy to fall into local optima. These weaknesses have restricted the wider applications of the ABC algorithm. To accelerate convergence speed and avoid the local optima, a number of variant ABC algorithms have been proposed [8-13]. However, so far, it is seen to be difficult to simultaneously achieve both goals.

To achieve the both goals, in this paper, we propose two novel solution search equations. In this method, the employed bees search only around the random solution to improve exploration, and the onlooker bees search only around the best solution to improve exploitation. We name the ABC algorithm using the novel solution search equations as ABCNSS algorithm.

The rest of this paper is organized as follows. In Section 2, the ABC algorithm is described. The improved ABC algorithm is proposed in Section 3. In Section 4, experimental results from several basic benchmark functions are shown and compare with the performance of basic ABC and five classical improved ABC algorithms. Finally, Section 5 gives conclusions.

2. Artificial bee colony algorithm

Artificial Bee Colony (ABC) algorithm is a biological-inspired optimization algorithm which simulates the foraging behavior of a bee colony. The artificial bee colony consists of three groups of bees: employed bees, onlooker bees and scout bees. In the ABC algorithm, the position of a food source represents a possible solution of the optimization problem, and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Furthermore, the number of employed bees or onlooker bees is set to be equal to the number of the solutions in the population.

In the very beginning, the initial population of solutions is filled with SN number of randomly generated D-dimensional real-valued vectors (i.e., food sources). Let $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ represent the i th solution in the population, and the x_i is generated as follows:

$$x_i^j = x_{\min}^j + rand(0,1)(x_{\max}^j - x_{\min}^j) \tag{1}$$

where $i = 1, 2, \cdots, SN$, $j \in \{1, 2, \cdots, D\}$. $x_{\min}^j$ and $x_{\max}^j$ are the lower and upper bounds for the dimension j, $rand(0,1)$ is a random number in the range $[0,1]$. These solutions are randomly assigned to SN number of employed bees and their fitness values are evaluated.

After initialization, each employed bee x_i generates a new food source v_i in the neighborhood of its present position by using search equation as follows:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where j is a random integer in the range $[1, D]$, k is a random integer in the range $[1, SN]$ and $k \neq i$, φ_{ij} is a random number in the range $[-1,1]$.

Once v_i is obtained, the fitness value is calculated for every v_i by

$$fitness_i = \begin{cases} \dfrac{1}{1+f_i}, & f_i \geq 0 \\ 1 + |f_i|, & f_i < 0 \end{cases} \tag{3}$$

where f_i is the function value of solution v_i. If the fitness value of v_i is equal to or better than that of x_i, v_i will replace x_i and become a new member of the population; otherwise x_i is retained

After all employed bees complete their searches, they share their information related to the nectar amounts and the positions of their sources with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source site with a probability related to its nectar amount. The probability of a food source being selected is calculated by

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{4}$$

where fit_i is the fitness value of the solution x_i, Obviously, the higher the fit_i is, the more probability that the i th food source is selected. An onlooker bee evaluates the nectar information taken from all the employed bees and selects a food source x_i depending on its probability value P_i. Once the onlooker has selected her food source x_i, she produces a modification on x_i by using (2). As in the case of the employed bees, if the modified food source has a better or equal nectar amount than x_i, the modified food source will replace x_i and become a new member in the population.

In ABC, if a food source x_i cannot be further improved through a predetermined number of $\lim it$, the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout bee. The scout bee produces a food source by using (1).

3. Artificial bee colony algorithm based on novel search strategy

According to the solution search equation of ABC algorithm described by Eq. (2), the coefficient φ_{ij} is a uniform random number in $[-1,1]$ and x_{kj} is a random individual in the population, therefore, the solution search dominated by Eq. (2) is random enough for exploration. In other words, the solution search equation described by Eq. (2) is good at exploration but poor at exploitation. In order to improve the exploitation, Zhu et al. [9] proposed an improved ABC algorithm called GABC by incorporating the information of global best solution into the solution search equation, the new solution is given by the following search equation:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) + \phi_{ij}(x_{best,j} - x_{ij}) \qquad (5)$$

where x_{best} is the global best solution in the population, ϕ_{ij} is a uniform random number in $[0, C]$,where C is a nonnegative constant. The modified solution search equation described by Eq. (5) can increase the exploitation of ABC algorithm.

Inspired by GABC, we propose two novel solution search equations as follows

$$v_{ij} = x_{rj} + \varphi_{ij}(x_{rj} - x_{kj}) + \phi_{ij}(x_{best,j} - x_{rj}) \qquad (6)$$

$$v_{ij} = x_{mj} + \varphi_{ij}(x_{mj} - x_{kj}) + \phi_{ij}(x_{best,j} - x_{mj}) \qquad (7)$$

where x_r is the solution which is randomly selected from the population, x_m is the solution which is selected from the population by using the roulette wheel, r and k are not equal to each other and i.

In Eq. (6), the random solution x_r is used to increase diversity in the population and increase the exploration, on the other hand, we takes advantages of the information of the global best solution to balance the exploration and exploitation. In this paper, Eq. (6) replaces Eq. (2) in the employed bee stage. In Eq. (7), the solution x_m which is selected from the population by using the roulette wheel is used to accelerate convergence speed, In this paper, Eq. (7) replaces Eq. (2) in the onlooker bee stage.

In this paper, we also focus the method that is used to compare and to select between the old solution and the new solution in each iteration. Basically, the comparison of the new solution and the old solution is done by the fitness value. If the fitness of the new solution is better than the fitness of the old solution, we select the new one and ignore the old solution. The fitness value can be obtained from Eq. (3).

Based on Eq. (3), we can see that when $f(x)$ is larger than the zero but has a very small value, e.g. 1E−20, the fitness value of equation 1/(1+1E−20) is rounded up to be 1 (1E−20 is ignored). This will lead the fitness of all solutions to become equal to 1 in the later iterations. In other words, there is no difference between the fitness values that is equal to 1/(1+1E−20) and 1/(1+1E−100). Thus, a new solution that gives a better fitness value than the old solution will be ignored and the solution will stagnate at the old solution. In order to solve this issue, we directly use the objective value of function for comparison and selection of the better solution.

4. Experiments

4.1. Benchmark functions and parameter settings

In this section, the ABCNSS is applied to minimize s set of 12 benchmark functions, those functions are presented in Table 1. Specifically, functions f_1 - f_5 are single peak functions; function f_6 is a noisy quartic function; functions f_7 - f_{12} are multimodal functions and the number of local minima increases exponentially with the problem dimension.

In order to testify the efficiency of the proposed algorithm, the experiment results are compared with the basic ABC algorithm. The population size of ABC and ABCNSS is 40 (i.e. $SN = 20$), all benchmark functions are tested on 30 dimensions, $\lim it$ is $SN * D$, the number of maximum function evaluations is set to 150000.

Table 1: Benchmark functions used in experiments

Function name	Function	Search range	Min
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$-100 \le x_i \le 100$	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$-10 \le x_i \le 10$	0
Schwefel 2.21	$f_3(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$	$-100 \le x_i \le 100$	0
Step	$f_4(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$-100 \le x_i \le 100$	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$-10 \le x_i \le 10$	0
Quartic	$f_6(x) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0,1)$	$-1.28 \le x_i \le 1.28$	0
Rastrigin	$f_7(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$-5.12 \le x_i \le 5.12$	0
Griewank	$f_8(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$-600 \le x_i \le 600$	0
Ackley	$f_9(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$	$-32 \le x_i \le 32$	0
Schaffer	$f_{10}(x) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^{D} x_i^2}\right) - 0.5}{\left(1 + 0.001\sum_{i=1}^{D} x_i^2\right)^2}$	$-100 \le x_i \le 100$	0
Penalized1	$f_{11}(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4) \quad y_i = 1 + \frac{1}{4}(x_i + 1),\ u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$-50 \le x_i \le 50$	0
Penalized2	$f_{12}(x) = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$-50 \le x_i \le 50$	0

4.2. Experimental results

The experimental results are shown in Table 2 in terms of the best, worst, mean and standard deviation (std) of the solutions obtained in the 30 independent runs by each algorithm. In order to compare the convergence rate of the ABC and ABCNSS, we draw the convergence curves of six tested functions in Fig.1.

Table 2: Performance comparisons of ABC and ABCNSS

Function	Algorithm	Best	Worst	Mean	Std
f_1	ABC	2.75E-16	6.95E-16	4.93E-16	7.98E-17
	ABCCSS	7.87E-126	2.84E-123	3.75E-124	6.00E-124
f_2	ABC	9.86E-16	1.63E-15	1.31E-15	1.54E-16
	ABCCSS	1.73E-65	1.29E-63	3.48E-64	3.56E-64
f_3	ABC	2.80E-01	2.39E+00	8.37E-01	4.72E-01
	ABCCSS	6.32E-03	2.08E-02	1.24E-02	3.84E-03
f_4	ABC	0	0	0	0
	ABCCSS	0	0	0	0
f_5	ABC	4.23E-04	1.74E-01	4.32E-02	4.51E-02
	ABCCSS	9.27E-04	7.11E+01	3.67E+00	1.30E+01

f_6	ABC	1.96E-02	7.11E-02	4.85E-02	1.29E-02
	ABCCSS	7.19E-03	1.77E-02	1.22E-02	2.96E-03
f_7	ABC	0	0	0	0
	ABCCSS	0	0	0	0
f_8	ABC	0	4.88E-06	1.78E-07	8.89E-07
	ABCNSS	0	0	0	0
f_9	ABC	2.55E-14	4.32E-14	3.55E-14	3.62E-15
	ABCNSS	1.48E-14	2.90E-14	2.08E-14	3.62E-15
f_{10}	ABC	2.28E-01	3.96E-01	3.18E-01	5.19E-02
	ABCNSS	1.27E-01	2.73E-01	2.23E-01	4.53E-02
f_{11}	ABC	3.31E-16	6.78E-16	4.93E-16	5.82E-17
	ABCNSS	1.57E-32	1.57E-32	1.57E-32	2.74E-48
f_{12}	ABC	4.12E-16	6.98E-16	5.26E-16	5.80E-17
	ABCNSS	1.35E-32	1.35E-32	1.35E-32	5.47E-48

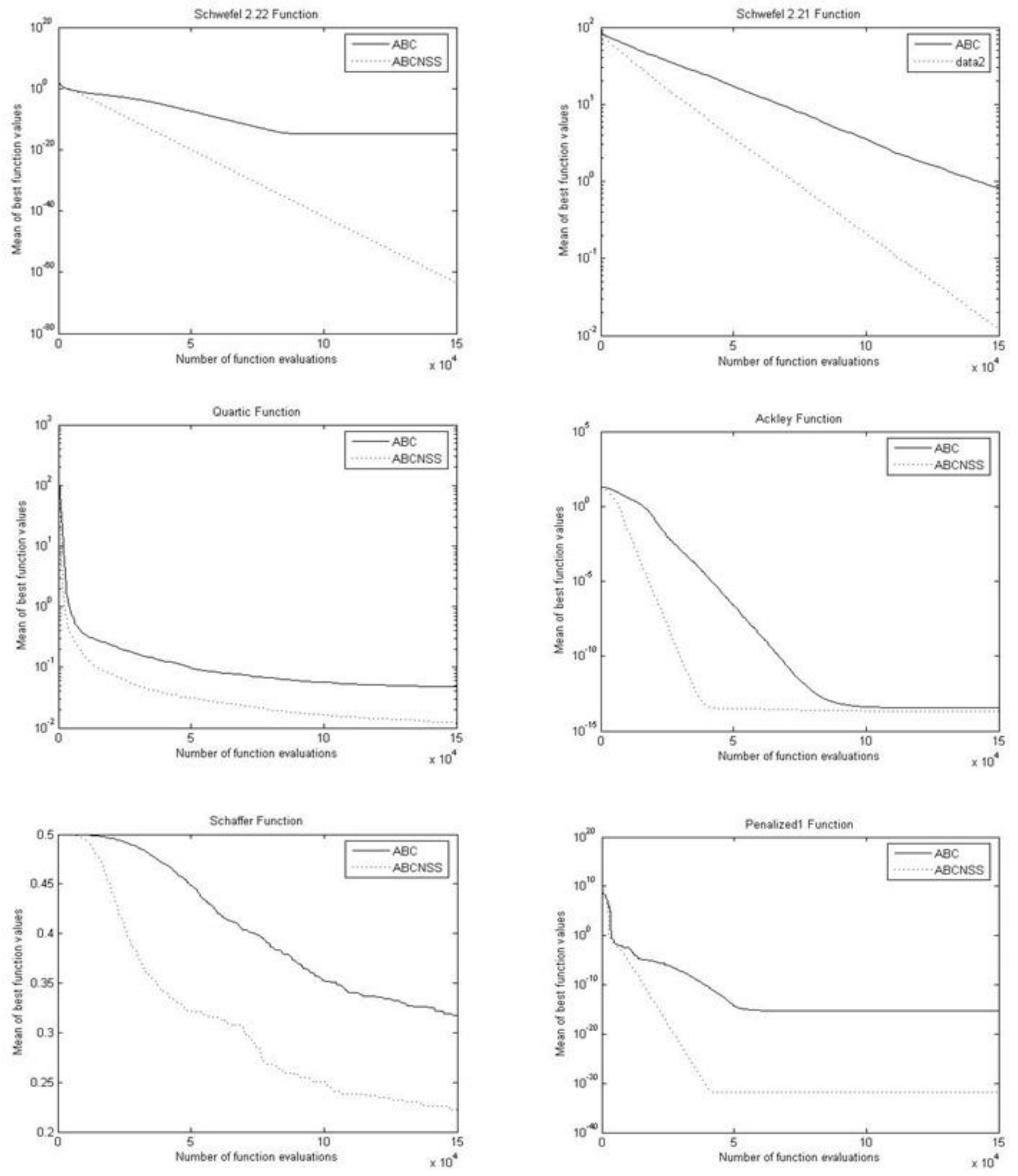

Figure 1: Convergence curves of ABC and ABCNSS on the 6 test functions

As is shown in Table 2, when solving the single peak functions, an interesting result is that two ABC-based algorithms have most reliably found the minimum of f_4. It is a region rather than a point in f_4 that is the optimum. Hence, this problem may relatively be easy to solve with a 100% success rate. For other single peak functions, ABCNSS offers the higher accuracy on almost all the functions except functions f_5. When solving the multimodal functions, ABCNSS can find the optimal or closer-to-optimal solutions on the complex functions f_7, f_8, f_9, f_{11} and f_{12}. As can be seen from Fig.1, Compare with ABC, ABCNSS has faster convergence speed and higher calculate accuracy. To sum up, the experimental results and Convergence curves clearly indicate that the ABCNSS is superior to ABC on almost all the functions.

In Table 3, ABCNSS is further compared with GABC [9], MABC [10], ABCBest1 [11], ABCBest2 [11] and ABCVSS [13], the number of maximum function evaluations is set to 150000. The best results are marked in bold. The results show that ABCNSS performs much better in most cases than these ABC variants. In a word, the ABCNSS algorithm can improve bees' searching abilities, prevent bees from falling into the local minimum, increase convergence speed and compute efficiency.

Table 3: Performance comparisons of GABC, MABC, ABCBest1, ABCBest2, ABCVSS and ABCNSS

Func.	GABC		MABC		ABCBest1		ABCBest2		ABCVSS		ABCCSS	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	4.62E-16	7.12E-17	9.43E-32	6.67E-32	3.11E-47	3.44E-47	5.96E-35	3.61E-35	1.53E-81	8.37E-81	3.75E-124	6.00E-124
f_2	1.35E-15	1.36E-16	2.40E-17	9.02E-18	2.10E-25	9.08E-26	1.36E-18	4.27E-19	7.89E-43	4.32E-42	3.48E-64	3.56E-64
f_3	2.18E-01	4.01E-02	1.02E+01	1.49E+00	2.18E+00	3.27E-01	3.55E+00	4.79E-01	4.08E-02	2.20E-02	1.24E-02	3.84E-03
f_4	0	0	0	0	0	0	0	0	0	0	0	0
f_5	3.21E-01	8.21E-01	6.11E-01	4.55E-01	1.49E+01	2.87E+01	5.45E+00	8.40E+00	3.87E-01	1.54E+00	3.67E+00	1.30E+01
f_6	2.03E-02	5.74E-03	3.71E-02	8.53E-03	2.06E-02	4.75E-03	2.53E-02	4.67E-03	1.81E-02	5.27E-03	1.22E-02	2.96E-03
f_7	0	0	0	0	0	0	0	0	0	0	0	0
f_8	3.70E-17	5.32E-17	0	0	0	0	1.81E-08	6.29E-08	0	0	0	0
f_9	3.20E-14	3.36E-15	4.13E-14	2.17E-15	3.01E-14	2.91E-15	3.07E-14	3.43E-15	2.45E-14	4.00E-15	2.08E-14	3.62E-15
f_{10}	2.66E-01	4.39E-02	2.95E-01	3.17E-02	2.39E-01	6.13E-02	2.81E-01	3.92E-02	2.84E-01	5.69E-02	2.23E-01	4.53E-02
f_{11}	4.12E-16	8.36E-17	1.90E-32	3.70E-33	1.57E-32	5.57E-48	1.57E-32	5.57E-48	1.57E-32	5.57E-48	1.57E-32	2.74E-48
f_{12}	4.01E-16	8.19E-17	2.23E-31	1.46E-31	1.35E-32	5.57E-48	1.35E-32	5.57E-48	1.35E-32	5.57E-48	1.35E-32	5.47E-48

5. Conclusion

In this paper, artificial bee colony algorithm based on novel search strategy was presented, called ABCNSS. In our method, the employed bees search only around the random solution to improve exploration, and the onlooker bees search only around the solution which is selected from the whole population depending on the roulette wheel to improve exploitation, at the same time, the comparison and the selection of the new solution were changed from a fitness-based comparison to an objective-value based. The performance of ABCNSS was compared with the basic ABC algorithm and several classical versions of ABC algorithm using a set of benchmark functions. The results demonstrate that the new algorithm outperforms the basic ABC algorithm and other variants of the ABC algorithm in terms of solution quality and robustness for most of the experiments.

References

[1]. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Kayseri, Turkey: Erciyes University; 2005.

[2]. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39(3), 2007, 171-459.

[3]. Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing, 8(1), 2008, 687-697.

[4]. Karaboga D, Basturk B. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 214(1), 2009, 108-132.

[5]. Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Applied Soft Computing, 9(2), 2009, 625-631.

[6]. D. Bose, S. Biswas, A.V. Vasilakos, S. Laha. Optimal filter design using an improved artificial bee colony algorithm. Information Sciences, 281(1), 2014, 443-461.

[7]. D. Karaboga, C. Ozturk. A novel clustering approach: artificial bee colony (ABC) algorithm. Applied Soft Computing, 11(1), 2011, 652-657.

[8]. F. Kang, J. Li, Ma Z Y. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. Information Sciences, 181(16), 2011, 3508-3531.

[9]. G. Zhu, S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation, 217(7), 2010, 3166-3173.

[10]. W. Gao, S. Liu. A modified artificial bee colony algorithm. Computer & Operations Research, 39(3), 2012, 687-697.

[11]. W. Gao, S. Liu, L. Huang. A global best artificial bee colony algorithm for global optimization. Journal of Computational and Applied Mathematics, 236(11), 2012, 2741-2753.

[12]. W. Gao, S. Liu, L. Huang. Enhancing artificial bee colony algorithm using more information-based search equations. Information Sciences, 270(1), 2014, 112–133.

[13]. Kiran M S, Hakli H, Gunduz M, *et al*. Artificial bee colony algorithm with variable search strategy for continuous optimization. Information Sciences, 300(1), 2015, 140-157.

Background

* This article is supported by The National Natural Science Foundation of China (Grant No. 11601234), the Key Project of JiangSu Natural Science Foundation (Grant No. BK20160571, BK20150537), he Key Project of JiangSu College Natural Science Foundation (15KJB110013).

Email-addresses: huimingxia1981@126.com